HAL
open science

Home Automation with a PLC: From requirement to commissionning

Jean-Paul Chemla, Bernard Riera

To cite this version:

Jean-Paul Chemla, Bernard Riera. Home Automation with a PLC: From requirement to commissionning. Informatika, 2017, 45, pp.12-18. hal-02990409

HAL Id: hal-02990409

https://hal.science/hal-02990409v1

Submitted on 17 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

HOME AUTOMATION WITH A PLC: FROM REQUIREMENT TO COMMISSIONNING

Jean-Paul Chemla, Bernard Riera
Université François Rabelais de Tours, Polytech Tours, France, CRESTIC, Université de Reims Champagne Ardenne, France

Keywords: PLC, Home Automation, Virtual home

1. ABSTRACT

This paper deals with the programming part of home automation. We will present a solution based on central PLC which controls all the basic devices in a home. The PLC is connected to buttons and sensors and can activate lights and shutters using relays. We will focus on the PLC program. We propose a methodology based on the definition and the use of a library of function blocks for each common device like buttons, lights and rollers. Once the library is defined, most of the programming task will consist in defining instances of the blocks, connecting them to define rules. The library should also take into account that there will be some human machine interface (HMI) to interact with the controlled home. Some solutions to build this kind of HMI are presented in the paper. In order to validate the controller, we propose to use virtual commissioning with a virtual house. In our case, we have used the software HOME I/O from Real Games, to perform this stage. At least, a real home with these solutions has been realized and comments about the use of this controlled home will be provided.

2. INTRODUCTION

2.1. Home automation

Home automation is a growing trend that is becoming more and more popular and affordable every day [1]. Today this term is rather replaced by smart house. It is used in electronics technology, automation, IT and telecommunications to ensure safety functions (alarms, cameras and remote monitoring), living comfort (automation and job scheduling daily, remote control, home care ...) and energy management (optimization of heating, lighting ...). The operating principle of a home automation system is to centralize the control and monitoring. Two different trends are on this business: wired solutions often expensive but offer a complete solution for a house, and home automation boxes which uses the power of Internet and wireless. With or without subscription, they allow open use and can be controlled from a computer, a smartphone or a touch pad. Installation is very simple, and takes just a few minutes by a non-expert user but it manages only a limited part of the house.

We consider here a global solution with a central PLC connected (often wired) to sensors, buttons and devices like lights, shutters, buzzers (for alarm) [2].

2.2. PLC and Home automation

Unlike smart devices, this solution can provide the possibility to interact with both classical buttons in the home and smartphones. Moreover, the PLC with his relays will be able to switch all kind of plain devices (lights, roller shutters) without buying specific ones. Even if the behavior is classical (just ON or OFF for a light), there will be no compatibility issues. The proposed solution is intended to be installed at the same time that the electrical wiring of the home because it is intended to control all home devices. As it is a non-computer based, industrial device, the PLC is more secured [3] and reliable than home automation boxes.

2.3. Context

The project will focus on basic devices of a home: lights, shutters and alarm. The main role of the PLC is to replace the classical wiring of the home to make it “smart”. Examples of behaviour will be provided in the programming part of this paper. The PLC has on his inputs:

- Buttons or switches (usually used for lights and shutters)
- Motion sensors in corridors
- Proximity sensor to detect if windows and doors are closed

The outputs of the PLC are connected to relays to activate:

- Lights
- Shutters
- Buzzers

It is possible to make a parallel with modern cars: the buttons and sensors are connected to a central controller which can switch on or off the headlights, lock or unlock the doors, go up or down the windows etc.

3. THE LIBRARY

As specified in the IEC 61131.3 norm, PLC can be programmed using programs, functions and function blocks. The environment provides built-in functions (as AND, XOR) and function blocks (as RTRIG, SR). The user can define his own programs, functions and function blocks in multiple languages (IL, LD, ST, SFC, FBD).

A large home can have lots of each device (buttons, lights…) and programming can be a repetitive task and difficult to maintain. Moreover, as we want a smart behaviour, each device can lead to a complex program. That’s why we will define our own function blocks for buttons, shutters, switches, lights, buzzers. These blocks will have input, output, input and output and private variables. They also will have a program part. Most of the programming software for PLC have the possibility to define new blocks. Let’s define some of these function blocks.

- Button:
  - Input: the button itself (an input of the PLC)
  - Outputs: short press, long press, double press

The button is simple: the function block detects if the button has been pressed shortly, pressed a long time (more than 1/2s) or doubled pressed. Most of our smartphones have a button with different functions according to these types of press.

- Light:
  - Inputs: switch (bool), off (bool), timer (bool), time for timer (time)
  - Outputs: the light itself (an output of the PLC)
  - Input-Outputs: HMI_switch, HMI_timer

We thought that a light can be switched, like traditional ones: from ON to OFF or vice versa. Sometimes, it could be convenient to turn the light ON for a limited time. Then it will turn OFF after that the time is up. In that case, we prevent 1 minute before time’s up by turn the light OFF for 5 seconds. If the timer input becomes true during the timer is running, the light will shut off except if it is in the last minute (in that case, the timer will be relaunched). For the HMI, it could be nice to have only 1 variable which tells if the light is ON or OFF and to control the light. That’s the role of the input-output variables: they are both information and control. HMI_switch is to switch the light, HMI_timer is for timer. For example, if the light is ON for a limited time, HMI_switch is OFF and HMI_timer is ON.

- Shutter:
  - Inputs: Up (bool), Down (bool), Stop (bool), length (time)
  - Outputs: Shutter Up (bool), Shutter Down (bool), Position (int)
  - Input-Outputs: Set point (int)

We define that if Up or Down inputs are ON shortly, the shutter moves until his end (up or down). If these inputs are ON for a long time (long press on the buttons) the shutter will stop when these inputs return to OFF (button is released). As there is no sensor of position (neither if they are opened or closed) in most of shutters, we have to define the time taken to completely open or close the shutter. Inside the block, the position of the shutter is calculated. This position is an output of the block as it can be useful for the HMI. Another value is useful for HMI: the set point. This variable is a percentage of closure. It is of course an input as the user can change it through his HMI. It is also an output of the block because we want that this value is set to the actual position when the shutter is not moving (this value is both an information and a control).

Other blocks have been defined for buzzers and switches for example.

Programming the block is the most technical part of the PLC programming because we have to:

- Anticipate a lot of situations that can happen. For example: the shutter is moving down, what happens if the down button is pressed? (in this case, we decided that the shutter has to stop). Other example: to stop the shutter, the user can press both buttons Up and Down. But the two inputs won't be synchronous.
- Be careful with all input and output variables because they can create instabilities. For example, if the user press a button to switch ON a light, the program sets ON the light and before that the HMI_switch is set to ON, the program can detect that the HMI_switch is in a different state from the light and light can return to OFF.

4. PROGRAMMING AND HMI

Once the library is defined, it remains:

- To declare all the inputs and output variables
- To create instances of the blocks for each button, light, shutter, etc…
- To create the variables for the communication with the HMI
- To create and manage variables for global actions
- To create an SFC to manage the alarm mode
- Then to program the behaviour of the house mainly by connecting the blocks between them and to other variables (global actions, HMI, alarm mode). This last part is not difficult and similar to "If that then that" rules programming but using blocks.

4.1. Programming

We give some details about some of the above points.

4.1.1. The alarm mode

We defined 5 different states of the alarm: "Presence mode", "Verifications before leaving", "Absent mode", "Waiting for digital code", "Intrusion". The SFC below gives the conditions of the transitions between states. The leaving condition can be for example a double press on a button near the door. When in the state "Verification before leaving", buzzer can beep quietly if every window and door are closed but long beeps can occur if it is not the case.

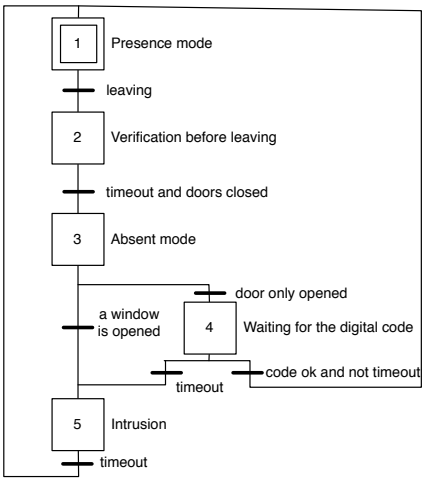

Figure 1.
SFC of the alarm mode

4.1.2. Examples of program using blocks

In the figure 2 below, there is the program equivalent to “If the button near the door is pressed for a short time OR if the door is opened AND there is no daylight AND the outdoor light is OFF THEN the outdoor light is ON for 15 minutes”. The global variable “global_outdoor_light_off” is calculated in an another part of the program. It can be set ON by multiple ways : the global light off of all the home or by a button on a HMI.

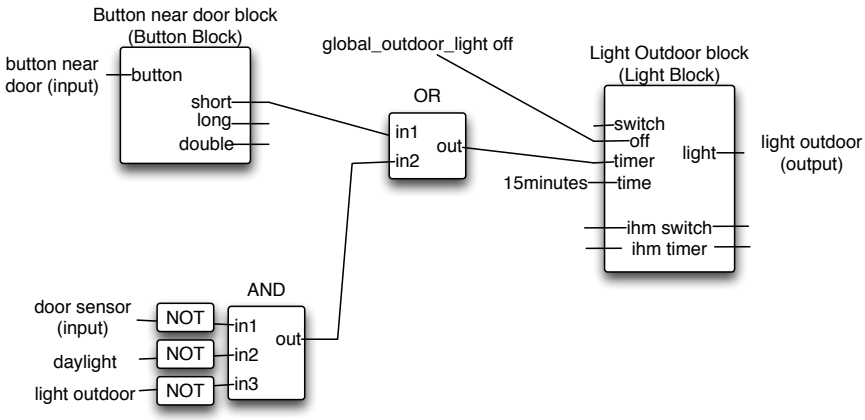

Figure 2
Example of Function block diagram program for a light

4.2. Human Machine Interfaces

As mentioned above, the human machine interface has been prepared by adding input/output variables in the definition of some blocks. In the program, variables have been created and localized to be accessible via the network with the Modbus TCP protocol. We have chosen this protocol because the PLC we used can manage it and as it uses TCP/IP, we can reach the PLC through different networks and 3G/4G. There is no need of an extra computer to have the interface (unlike [4]). The interface itself has to be easy to use and has to give a very quick access to controls. A smartphone has these qualities.

In the Apple App store or in the Google Play store, we can find applications which can connect to PLC using industrial network protocols and we can create interfaces to interact with the PLC. For example: mySCADA (both stores) can use multiple protocols. Other apps are specific to a brand of PLC. We have

chosen HMI Draw from SweetWilliams [6] in the Apple App store because it is ergonomic and has a large range of possibilities like using sensors of the device. In the figure 3 below, there is a view we can obtain with HMI Draw. In this view, we can see a switch for a light.

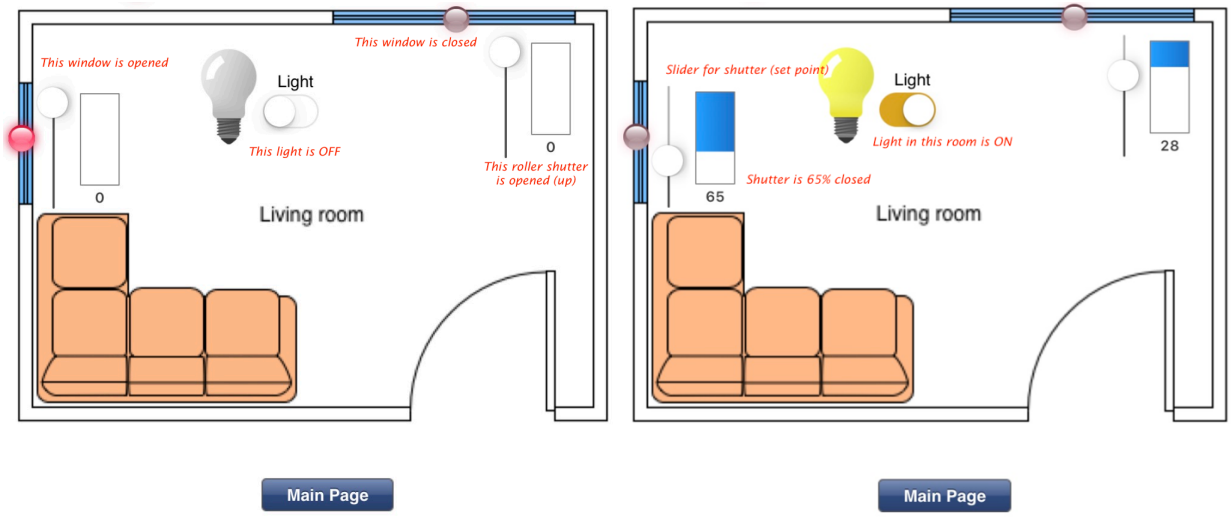

Figure 3
Example of a view with the App HMI Draw

Here are the different steps to obtain this switch which can control a light and also indicate if the light is On or Off :

- In the PLC, the input and output variable HMI_switch of the light block is connected to a localized variable.
- In HMI Draw we declare a PLC Connector (like in the figure 4.1) to connect to the PLC.
- With this PLC connector, we declare a tag (Boolean) with the correct address to access the variable in the PLC (like in the figure 4.2, C197 if for %M196 of the PLC)
- In a page, we create a switch, and in the value item we put the tag we just declared (this is useful to indicate if the tag is true or false). See figure 4.3
- We return to the tag configuration to add the value of the switch in the “write” item of the tag. This is useful to write the tag when the switch is touched by the user.

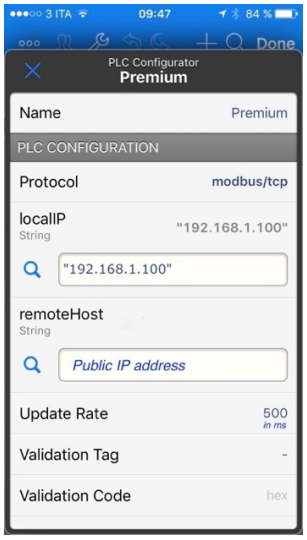

Figure 4.1

PLC configuration

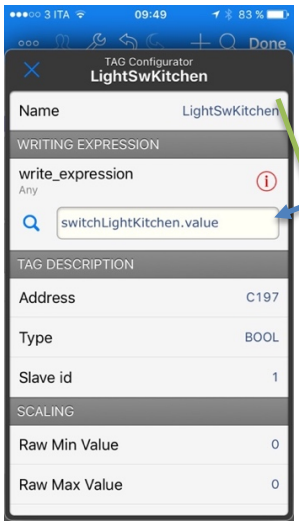

Figure 4.2

Tag configuration

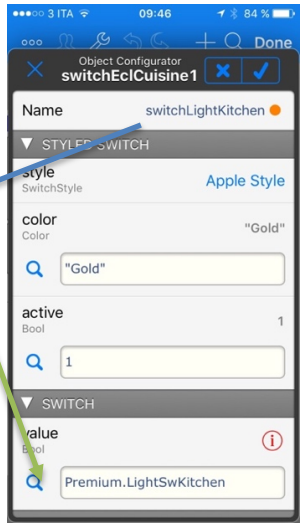

Figure 4.3

Switch configuration

5. VIRTUAL COMMISSIONNING

In order to test the program of the PLC, we looked for a solution with a virtual home which can be connected to a real PLC. This solution is the software called HOME I/O [7]. This software is much more a serious game than a simulation. It is the result of a 3-year R&D project “DOMUS” (2011-2014) partially founded by the French Ministry of National Education, between CReSTIC lab from the University of Reims Champagne-Ardenne and Real Games, a Portuguese company [5]. The idea has been to bring a virtual house into the class room, usable from middle schools to universities, adapted to learners (Y generation) and teachers and suitable for control and STEM (Control, Science, Technology, Engineering and Math) education. HOME I/O has been designed by applying a systemic approach of education. For that, HOME I/O enables to study the house from different points of view (automation, energy efficiency, smart home …), as a whole or as a set of subsystems, but taking into account the environment. The possibilities of pedagogical scenarios (i.e. situations, questions, problems) offered by HOME I/O depend only on teachers’ imagination. HOME I/O seems to be a very innovative pedagogical tool and it was awarded with the prize “Le Coup de Coeur” in the 6th Edition of “TROPHÉES DES TECHNOLOGIES ÉDUCATIVES 2014” at EDUCATEC-EDUCATICE in Paris.

Hence, HOME I/O is real time simulation software (Fig. 5) of a smart house and its surrounding environment, designed to cover a wide range of curriculum targets within Control, Science, Technology, Engineering and Math, from middle schools to universities.

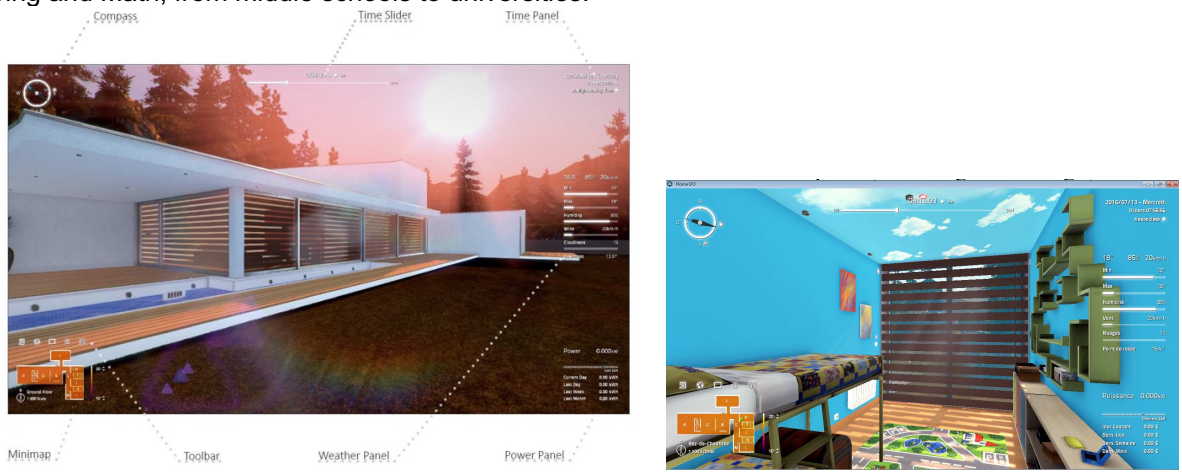

Figure 5
HOME I/O head-up display (left) and view of the kid room (right)

This virtual house, like a video game in first person (FPS, first person shooter), becomes a place of discovery and experimentation. HOME I/O is a realistic modern virtual house. It is necessary to make choices in equipment of the house and the possibilities of automation. It is possible to control lighting, rollers, electric heating, an alarm system for intrusion safety and home safety (fire). Altogether HOME I/O offers 174 interactive objects (lights, switches, inverters, roller blinds, gate, garage door, sirens, motion detectors, lighting, fire detectors, door opening sensor, remote control, radiators, thermostats) that can be used to automate the home or as a source of logical or numerical data (Fig. 6).

Figure 6
Interactive objects in HOME I/O

With another software, CONNECT IO (from the same company) this virtual home can be controlled by an external device like a PLC, using communication protocol (like Modbus TCP) or using a DAQ card with wired connections. This virtual commissioning gave us the opportunity to test function blocks, program and HMI interfaces.

6. REAL HOME

One of the author has a PLC in his home and applied the methodology given in this paper. This experience leads to these remarks:

- There is two kind of wires in the house: for control, with thin wires and for power with classical copper wires. Both of them are connected to the PLC. We could have chosen a network for the control wiring. For example, KNX is often used in that cases and lots of PLC can manage this network.
- If someone has a classical behaviour in the house, the actuators will also have a classical behaviour. Except the proximity sensors in the corridors and the timers for some lights, it seems to be a classical house.
- Global control for shutters (with a long press on a shutter button) is very convenient
- It is possible to switch on multiple lights with 1 button.
- We can add easily new features when we think about it by modifying the program section without creating new variables and without changing the function blocks. For example, turning on a light behind a door when opened like in figure 2 was done by adding the OR and AND blocks.
- The alarms part of the program is mainly used to switch off all lights and to verify that all windows are closed. The PLC also sends email when someone remove the alarm which is convenient to know when children come back from school.
- The PLC is running since more than 5 years without major trouble. Minor ones have been experienced at the beginning. We made some adjustments in function blocks and since years, we didn't change our library even if the program itself has changed many times.
- We changed the app for the HMI independently to the program.
- Even if it is not very difficult to change rules, only people who are able to program a PLC are able to modify them.

7. CONCLUSION

We presented a methodology to program a central PLC which controls all standard devices in a home. The library we have developed is based on standard languages and can be implemented in quite every brand of PLC. We use a Scada application to provide a convenient interface. The original point of our paper is the commissioning on both a simulated and a real home. The behaviour of the programmed home offers functionalities like global controls, automated lights ON or OFF, interface with a smartphone, emails on events etc... To improve our work, we will focus on the possibility to add connected objects to our central PLC. One way to make this connection should be through an home automation boxes which can communicate with the PLC.

REFERENCES

[1] DC Khedekar and al. *Home Automation-A Fast - Expanding Market*. Thunderbird International Business Review. 59, 1, 79-91, Jan. 2017. ISSN: 10964762

[2] J. Figueiredo, J.S. da Costa, *Predictive control applied to builing automation : an experimental setup,* IFAC Proceedings Volumes, 2006

[3] A. Jacobson, M. Boldt, B. Carlsson, *A risk analysis of a smart home automation system,* In Future Generation Computer Systems March 2016 56:719-733

[4] Okan Bingol, Kubilay Tasdelen *Web-based Smart Home Automation: PLC-controlled Implementation.,* Acta Polytechnica Hungarica. 3, 51, 2014. ISSN: 2064-2687

[5] Riera, B. Emprin, F., Annevicque, D., Colas, M., Vigario, B. *HOME I/O: a virtual house for control and STEM education from middle schools to Universities*. 11th IFAC Symposium on Advances in Control Education ACE 2016 — Bratislava, Slovakia, 1—3 June 2016

Internet references (last download: 31. 05. 2017):

[6] HMI Draw home page: http://www.sweetwilliamsl.com
[7] Home I/O Home page: https://realgames.co/home-io/